

WHAT IS CLAIMED IS:

1. A method of displaying compiler optimized source code, the method comprising:
generating an optimized source code for an original source code; and
displaying the optimized source code on an output device to visually indicate a change performed to the original source code in accordance to a compiler optimization.
2. The method of claim 1 wherein generating is performed in response to a request containing a type of the compiler optimization to be applied on the original source code.
3. The method of claim 1 wherein the optimized source code comprises a decompiled version of an object code generated from the original source code.
4. The method of claim 1 wherein the original source code is compiled according to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination.
5. The method of claim 4 wherein displaying comprises:
displaying a number of times a procedure call in the original source code is inlined.
6. The method of claim 1 wherein the original source code comprises a bytecode generated using a Java compiler.
7. The method of claim 6 further comprising:
displaying a number of times a procedure in the original source code is executed by an interpreter of a virtual machine program before being compiled by a

run-time compiler of the visual machine program.

8. The method of claim 1 wherein the optimized source code and the original source code are simultaneously displayed in separate windows of a user interface on the output device.

9. The method of claim 1 wherein displaying comprises:
identifying a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization on the original source code to generate the optimized source code; and
highlighting the differences on a user interface on the output device.

10. The method of claim 1 wherein displaying comprises:
identifying a failed optimization on the original source code;
determining a reason for the failed optimization from a compiler used to optimize the source code; and
displaying the reason for the failed optimization.

11. The method of claim 10 wherein the identifying is performed in response to a user query.

12. A method of displaying compiler optimized source code, comprising:
generating an object code from an original source code;
optimizing the object code to produce an optimized object code;
decompiling the optimized object code to produce an optimized source code; and
simultaneously displaying the optimized source code and the original source code in separate windows of a user interface on an output device to visually indicate a change to the original source code as a result of the optimizing.

13. The method of claim 12 wherein generating is performed in response to a request containing a type of a compiler optimization to be applied on the original source code to generate the optimized source code.

14. The method of claim 12 wherein the original source code is compiled according to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination.

15. The method of claim 12 wherein simultaneously displaying comprises:
identifying a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization on the original source code; and
highlighting the differences on a user interface on the output device.

16. An apparatus for displaying compiler optimized source code, the apparatus comprising:
a memory for storing a program editor, a compiler program and a decompiler program;
an output device for displaying an optimized source code; and
a processor, for executing the program editor, the compiler program and the decompiler program, the processor being configured to:
generate the optimized source code for an original source code, and
provide, to the output device, the optimized source code to visually indicate a change performed to the original source code in accordance to a compiler optimization of the compiler program.

17. The apparatus of claim 16 wherein the optimized source code is generated in response to a request containing a type of a compiler optimization to be applied on the original source code.

18. The apparatus of claim 16 wherein the compiler program is used to compile the original source code in accordance to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination.

19. The apparatus of claim 16 wherein the processor is further configured to:
identify a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization on the original source code; and
highlight the differences on a user interface on the output device.

20. A computer readable medium storing a software program that, when executed by a processor of a computer, causes the computer to perform operations comprising:
generating an optimized source code for an original source code using a compiler; and
displaying the optimized source code on an output device to visually indicate a change performed to the source code using a compiler optimization.

21. The computer readable medium of claim 20 wherein the compiler is configured to perform only the compiler optimization selected by a user.

22. The computer readable medium of claim 20 wherein generating is performed in response to a request containing a type of the compiler optimization to be applied on the original source code.

23. The computer readable medium of claim 20 wherein the optimized source code comprises a decompiled version of an object code generated from the original source code.

24. The computer readable medium of claim 20 wherein the original source

code is compiled according to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination.

25. The computer readable medium of claim 24 wherein displaying comprises:
displaying a number of times a procedure call in the original source code is inlined.

26. The computer readable medium of claim 20 wherein the original source code comprises a bytecode generated using a Java compiler.

27. The computer readable medium of claim 26 further comprising:
displaying a number of times a procedure in the original source code is executed by an interpreter of a visual machine program before being compiled by a run-time compiler of the visual machine program.

28. The computer readable medium of claim 20 wherein the optimized source code and the original source code are simultaneously displayed in separate windows of a user interface on the output device.

29. The computer readable medium of claim 20 wherein displaying comprises:
identifying a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization on the original source code to generate the optimized source code; and
highlighting the differences on a user interface on the output device.

30. The computer readable medium of claim 20 wherein displaying comprises:
identifying a failed optimization on the original source code;
determining a reason for the failed optimization from a compiler used to optimize the source code; and
displaying the reason for the failed optimization.

31. The computer readable medium of claim 20 wherein the identifying is performed in response to a user query.

32. A computer readable medium storing a software program that, when executed by a processor of a computer, causes the computer to perform operations comprising:

generating an object code from an original source code using a compiler;
optimizing the object code to produce an optimized object code;
decompiling the optimized object code to produce an optimized source code; and
simultaneously displaying the optimized source code and the original source code in separate windows of a user interface on an output device to visually indicate a change to the original source code as a result of the optimizing.

33. The computer readable medium of claim 32 wherein the compiler is configured to perform only the compiler optimization selected by a user.

34. The computer readable medium of claim 32 wherein generating is performed in response to a request containing a type of a compiler optimization to be applied on the original source code to generate the optimized source code.

35. The computer readable medium of claim 32 wherein the original source code is compiled according to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination.

36. The computer readable medium of claim 32 wherein simultaneously displaying comprises:

identifying a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization on the

original source code; and

highlighting the differences on a user interface on the output device.

09917958-07304
T00E20"856ZT660